

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-312189

(43)Date of publication of application : 24.11.1998

(51)Int.Cl. G10H 1/02
G10H 1/00
G10H 1/00
G10H 7/02

(21)Application number : 10-075037

(71)Applicant : YAMAHA CORP

(22)Date of filing : 10.03.1998

(72)Inventor : TAMURA GENICHI

(30)Priority

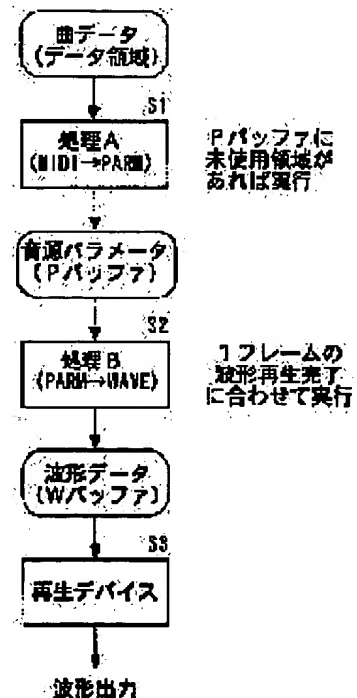
Priority number : 09 72845 Priority date : 11.03.1997 Priority country : JP

(54) MUSICAL SOUND GENERATION METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a musical sound generation method capable of decentralizing and reducing a processing load by hourly averaging the processing load for generating the waveform data from the music data.

SOLUTION: The music data includes the performance event data and durative information. The processing A of S1 develops the music data to sound source parameters to write them in a P buffer. This S1 is executed when an unused area exists in the P buffer. The processing B of S2 inputs the sound source parameters added with time information of a time frame (F) in a time frame (F-1) whenever reproduction of one time frame (e.g. F-2) is ended to write them in a W buffer. A DMAC(direct memory access controller) controlled by a CODEC driver reads out the waveform data stored in the W buffer one sample each at every sampling period to output a waveform through a sound system.



LEGAL STATUS

[Date of request for examination] 16.03.1999

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2000 Japanese Patent Office

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 10 - 3 1 2 1 8 9

(43) 公開日 平成10年(1998)11月24日

(51) Int. Cl. °

識別記号

F I

G 1 0 H 1/02
1/00

1 0 2

7/02

G 1 0 H 1/02
1/00

Z

1 0 2 Z

7/00 5 2 1 Z

審査請求 未請求 請求項の数 4

F D

(全 1 4 頁)

(21) 出願番号 特願平10-75037

(22) 出願日 平成10年(1998)3月10日

(31) 優先権主張番号 特願平9-72845

(32) 優先日 平9(1997)3月11日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000004075

ヤマハ株式会社

静岡県浜松市中沢町10番1号

(72) 発明者 田邑 元一

静岡県浜松市中沢町10番1号 ヤマハ株式
会社内

(74) 代理人 弁理士 浅見 保男 (外2名)

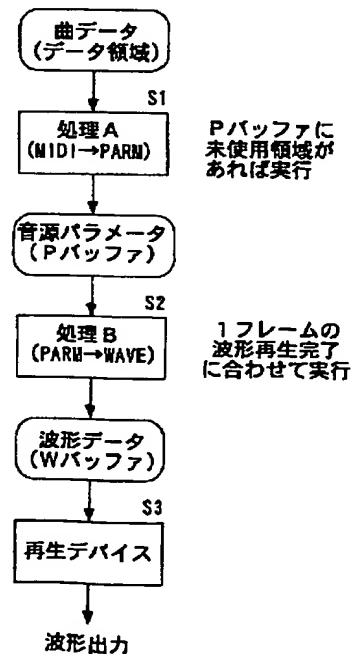
(54) 【発明の名称】 楽音発生方法

(57) 【要約】

【課題】 負荷が分散され処理負担が小さくなる楽音発生方法を提供する。

【解決手段】 曲データには、演奏イベントのデータとデュレーション情報が含まれている。S1の処理Aは、曲データを音源パラメータに展開しPバッファに書き込む。このS1は、Pバッファに未使用領域がある場合に実行される。S2の処理Bは、1つの時間フレーム（例えば、F-2）の再生が終了する毎に、時間フレーム（F-1）において、時間フレーム（F）の時刻情報が付与された音源パラメータを音源レジスタに入力し、時間フレーム（F）で再生する波形データを生成し、Wバッファに書き込む。CODECドライバに制御されたD MACが、Wバッファに記憶された波形データをサンプリング周期毎に1サンプルずつ読み出し、サウンドシステムを通して波形を出力する。

曲ファイルの再生



【特許請求の範囲】

【請求項1】 供給された曲データを順次パラメータに変換し第1のメモリに書き込む第1のステップと、第1のメモリに記憶された前記パラメータに基づいて波形データを生成し第2のメモリに書き込むとともに、前記波形データの生成に使用した前記パラメータが書き込まれていた領域を第1のメモリにおいて解放する第2のステップと、第2のメモリに記憶された前記波形データを再生して楽音を発生する第3のステップを有し、第2のステップは、第3のステップの前記波形データの再生の進行に応じて前記波形データを生成し、第1のステップは、第1のメモリに未使用領域が存在する場合に前記パラメータへの変換を実行することを特徴とする楽音発生方法。

【請求項2】 通信路を通じて供給された曲データを受信し第1のメモリに書き込む第1のステップと、第1のメモリに記憶された前記曲データをパラメータに変換し第2のメモリに書き込む第2のステップと、第2のメモリに記憶された前記パラメータに基づき波形データを生成し第3のメモリに書き込むとともに、前記波形データの生成に使用した前記パラメータが書き込まれていた領域を第2のメモリにおいて解放する第3のステップと、第3のメモリに記憶された前記波形データを再生して楽音を発生する第4のステップを有し、第3のステップは、第4のステップの前記波形データの再生の進行に応じて前記波形データを生成し、第2のステップは、第2のメモリに未使用領域が存在し、かつ、第1のメモリに未だパラメータに変換されていない曲データが存在する場合に前記パラメータへの変換を実行することを特徴とする楽音発生方法。

【請求項3】 曲データを供給する供給ステップと、供給された前記曲データから、演奏イベントの演奏データおよび演奏タイミングを取り出し、前記演奏データを音源パラメータに変換し前記演奏タイミングとともにメモリに書き込むパラメータ変換ステップと、所定時間間隔ごとに起動され、音源レジスタに記憶された前記音源パラメータに基づいて楽音波形を生成するステップであり、前記メモリに前記音源パラメータとともに記憶された前記演奏タイミングが、前記楽音波形を生成しようとする時刻に一致する時点で、前記演奏タイミングに対応する前記音源パラメータにより前記音源レジスタの内容を更新しつつ、前記音源レジスタに記憶された前記音源パラメータに基づいて前記楽音波形の生成を実行する楽音波形生成ステップと、生成された前記楽音波形を前記演奏タイミングに対応させて再生する楽音発生ステップと、を備えており、前記パラメータ変換ステップは、処理の優先順位が前記楽音波形生成ステップよりも低く設定され、かつ、前記楽音波形生成ステップとは独立して先行して実行できるようにしたことを特徴とする楽音発生方法。

【請求項4】 前記パラメータ変換ステップが実行されていない前記演奏イベントの前記演奏タイミングが、前記楽音波形生成ステップが前記楽音波形を生成しようとする前記所定時間間隔の範囲内にあるときには、前記楽音波形生成ステップが前記楽音波形を生成する前に、前記演奏イベントに対して前記パラメータ変換ステップを実行することを特徴とする請求項3に記載の楽音発生方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、MIDIデータなどの曲データの供給を受けて楽音を発生する楽音発生方法に関するものである。例えば、ネットワークを経由してMIDIデータストリームを受信しながら再生するソフト音源に適用することができるものである。

【0002】

【従来の技術】特別なハードウェアを用いずにソフトウェアにより音源を実現するソフト音源が知られている。このようなソフト音源を用いて、あらかじめ用意された曲データを再生する場合があるが、最近では、ネットワークから曲データをロードして再生する場合もある。

【0003】コンピュータから、インターネット等のネットワークを通じて曲データを要求する命令をサーバに送り、サーバは、この命令に回答して曲データをコンピュータに向けて配送する。具体的な例としては、ワールドワイドウェブ(WWW)のホームページにハイパーテキスト言語(HTML)に基づくタグとして埋め込まれた曲データを、コンピュータが、このホームページのロード時に要求する場合や、コンピュータからファイル転送プロトコル(FTP)に基づくFTPサーバ上に用意されたMIDIファイルを、画面上の特定部分をクリックすることにより要求する場合等がある。曲データとしては、スタンダードMIDIファイル(SMF)、および、このSMFに画像データや歌詞データを組み合わせたカラオケファイルなどのマルチメディアファイルがある。

【0004】コンピュータでは、曲データを受信するため、ウェブブラウザ等のソフトウェアが実行される。ウェブブラウザとしては、ネットスケープ(Netscape Communications Corporation、商標)やインターネットエクスプローラ(Microsoft Corporation、商標)が一般的である。ウェブブラウザでは、上述した手法により曲データのファイルを、サーバからロードする機能を有する。

【0005】その具体例として、「MIDIPLUG」(ヤマハ株式会社、商標)「Crescendo」(Live Up Date Corporation、商標)、「歌楽」(ヤマハ株式会社、商標)などがある。

50 「MIDIPLUG」は、XG規格の主要音色を備え、

MIDIデータを完全に受信し終わると再生を開始するソフト音源である。「Crescendo」は、MIDIデータを受信しながら、受信データがある程度たまるとローカルのハードまたはソフトによる外部MIDI音源にMIDI出力を行うプレーヤである。これらは、ワールドワイドウェブ(WWW)のホームページを閲覧するウェブブラウザに組み込まれるプラグインソフトであり、別ソフトウェアによるウェブブラウザへの付加機能となっているが、将来的には、ウェブブラウザ自身の標準機能となる可能性もある。「歌楽」は、歌詞と演奏とを同期させて表示する通信カラオケ用ソフトウェアパッケージである。

【0006】図14は、従来の楽音発生方法の動作説明図である。図中、121はMIDIデータの入力タイミング、12は音源パラメータ作成期間、14は波形データ生成期間である。従来のソフト音源を用いた楽音発生方法において、音源ドライバは、時間フレーム(F-2)の期間のMIDIデータ入力タイミング121でMIDIデータを入力し、音源パラメータ作成期間12において音源パラメータの生成を行い、次の時間フレーム(F-1)の期間の波形データ生成期間14において、新たに生成された音源パラメータを含めた波形データを生成する。この波形データは、次の時間フレーム(F)において再生される。音源パラメータの生成は、MIDIデータが入力されるごとに、波形データの生成を中断して行われる。

【0007】そのため、ノートオン、ノートオフ、ピッチベンド等の演奏イベントが特定の時間フレーム、例えば、時間フレーム(F-2)に集中した場合、この時間フレームでCPUの負荷が極端に増大し、音源計算に影響を与え、同時発音数の減少など波形の生成が完全に行えなくなるという問題があった。

【0008】

【発明が解決しようとする課題】本発明は、上述した問題を解決するためになされたもので、曲データから波形データを生成する処理の負荷を時間的に平均化することにより実質的な処理負担が小さくなる楽音発生方法を提供することを目的とするものである。また、ネットワークから曲データをロードして波形データを再生する場合に、曲データの全部のロードが完了するのを待つことなく、曲の再生を開始することができる楽音発生方法を提供することを目的とするものである。

【0009】

【課題を解決するための手段】請求項1に記載の発明においては、楽音発生方法において、供給された曲データを順次パラメータに変換し第1のメモリに書き込む第1のステップと、第1のメモリに記憶された前記パラメータに基づいて波形データを生成し第2のメモリに書き込むとともに、前記波形データの生成に使用した前記パラメータが書き込まれていた領域を第1のメモリにおいて

解放する第2のステップと、第2のメモリに記憶された前記波形データを再生して楽音を発生する第3のステップを有し、第2のステップは、第3のステップの前記波形データの再生の進行に応じて前記波形データを生成し、第1のステップは、第1のメモリに未使用領域が存在する場合に前記パラメータへの変換を実行するものである。

【0010】したがって、曲データから波形データを生成する処理を、波形データの再生の進行に応じて処理をする部分と、これとは独立して実行することのできるパラメータ変換処理とに分けることができたため、パラメータ変換処理を、この処理のためのバッファ用である第1のメモリの記憶容量の範囲で先行して実行することができ、曲データから波形データを再生する処理の負荷が時間的に平均化され、実質的な処理負担が小さくなる。

【0011】請求項2に記載の発明においては、楽音発生方法において、通信路を通じて供給された曲データを受信し第1のメモリに書き込む第1のステップと、第1のメモリに記憶された前記曲データをパラメータに変換し第2のメモリに書き込む第2のステップと、第2のメモリに記憶された前記パラメータに基づき波形データを生成し第3のメモリに書き込むとともに、前記波形データの生成に使用した前記パラメータが書き込まれていた領域を第2のメモリにおいて解放する第3のステップと、第3のメモリに記憶された前記波形データを再生して楽音を発生する第4のステップを有し、第3のステップは、第4のステップの前記波形データの再生の進行に応じて前記波形データを生成し、第2のステップは、第2のメモリに未使用領域が存在し、かつ、第1のメモリに未だパラメータに変換されていない曲データが存在する場合に前記パラメータへの変換を実行するものである。

【0012】したがって、請求項1に記載の発明と同様な作用を奏するとともに、第1のメモリに未だパラメータに変換されていない曲データが存在する場合に、パラメータ変換処理を行うことができるため、曲データの全部のロードが完了するのを待つことなく、曲の再生を開始することができる。

【0013】請求項3に記載の発明においては、楽音発生方法において、曲データを供給する供給ステップと、供給された前記曲データから、演奏イベントの演奏データおよび演奏タイミングを取り出し、前記演奏データを音源パラメータに変換し前記演奏タイミングとともにメモリに書き込むパラメータ変換ステップと、所定時間間隔ごとに起動され、音源レジスタに記憶された前記音源パラメータに基づいて楽音波形を生成するステップであり、前記メモリに前記音源パラメータとともに記憶された前記演奏タイミングが、前記楽音波形を生成しようとする時刻に一致する時点で、前記演奏タイミングに対応する前記音源パラメータにより前記音源レジスタの内容

を更新しつつ、前記音源レジスタに記憶された前記音源パラメータに基づいて前記楽音波形の生成を実行する楽音波形生成ステップと、生成された前記楽音波形を前記演奏タイミングに対応させて再生する楽音発生ステップと、を備えており、前記パラメータ変換ステップは、処理の優先順位が前記楽音波形生成ステップよりも低く設定され、かつ、前記楽音波形生成ステップとは独立して先行して実行できるようにしたものである。

【0014】したがって、曲データから波形データを生成する処理を、波形データの再生の進行に応じて処理をする部分と、これとは独立して実行することのできるパラメータ変換処理とに分けることができたため、パラメータ変換処理を、先行して実行することができ、曲データから波形データを再生する処理の負荷が時間的に平均化され、実質的な処理負担が小さくなる。

【0015】請求項4に記載の発明においては、請求項3記載の楽音発生方法において、前記パラメータ変換ステップが実行されていない前記演奏イベントの前記演奏タイミングが、前記楽音波形生成ステップが前記楽音波形を生成しようとする前記所定時間間隔の範囲内にあるときには、前記楽音波形生成ステップが前記楽音波形を生成する前に、前記演奏イベントに対して前記パラメータ変換ステップを実行するものである。

【0016】したがって、楽音波形を生成しようとする所定時間間隔の範囲内にパラメータ変換ステップがまだ実行されていない演奏イベントがある場合には、パラメータ変換ステップを実行してから楽音波形生成ステップを実行することができる。

【0017】

【発明の実施の形態】図1は、本発明の第1の実施の形態の楽音発生方法の処理の流れを表わす説明図である。図2は、本発明の第1の実施の形態の楽音発生方法のタイミング図である。図中、図14と同様な部分には同じ符号を付して説明を省略する。11は曲データ、13は演奏タイミングである。この実施の形態は、ローカルにあらかじめ用意された自動演奏用の曲ファイルの曲データを再生するものである。

【0018】図2に示す曲データ11は、データ領域に記憶されている。この曲データ11には、1つの演奏イベントにつき、この演奏イベントを表すデータとこの演奏イベントの演奏タイミングのデータが含まれている。演奏タイミングのデータは、曲の先頭などからの絶対時間を表す時刻情報でもよいし、(直前の演奏イベントから今回の演奏イベントまでの相対な時間間隔を表すデュレーション情報でもよい。異なるシーケンサ間で演奏データをやりとりするための標準ファイルとして、SMF(スタンダードMIDIファイル)が知られている。このSMFでは、デュレーション情報を用いて演奏タイミングを表している。以下、デュレーション情報を用いて演奏タイミングを表わす場合について説明する。

【0019】S1の処理Aにおいては、データ領域から曲データ11(MIDI)を順次読み出し、これを音源パラメータ(PARM)に展開しPバッファ(パラメータバッファ)に書き込む。ただし、S1は、Pバッファに未使用領域(空き)がある場合に実行し、あらかじめ音源パラメータ(PARM)をPバッファに書き込んでおく。この実行は、通常、図2に示すように実際の演奏タイミング13よりも前になされている。なお、図2では、演奏タイミング13は、図14に示した従来の楽音発生方法の動作説明図に合わせて、実際に波形データが再生される時間フレームの2フレーム前に図示している。

【0020】パラメータの1例としては、PCM音源の場合、選択された音色データに対応した波形メモリのスタートアドレスやエンドアドレス、エンベロープのパラメータなどがある。上述したデュレーション情報は、図2に示した演奏タイミング13を示す時刻情報データに変換され、Pバッファには、この時刻情報を付加した音源パラメータが書き込まれる。しかし、何らかの時間間隔情報(曲データ11中に含まれたデュレーション情報そのものでもよい)を付加して音源パラメータを書き込むようにしてもよい。いずれにせよ、時刻情報または時間間隔情報といった演奏タイミングを示す時間情報付きで音源パラメータが書き込まれ、波形生成の際に演奏タイミングを知るのに用いられる。

【0021】S2の処理Bは、1つの時間フレームの波形再生完了に合わせて実行され、図2に示すような時間フレームの開始時点ごとに起動がかけられる。この実施例では、1つの時間フレーム(例えば、F-2)の再生が終了する毎に、次に再生する時間フレームのさらにその次の時間フレーム(F)で再生すべき波形データを生成するようになっている。音源パラメータに付与された時刻情報がその波形データを生成する時間フレームの範囲に入っていれば、この音源パラメータは、波形データの生成の途中で音源レジスタに書き込まなければいけない。つまり、その波形データを再生する時間フレーム(F)の1つ前の時間フレーム(F-1)において、その時間フレーム(F)の時刻情報が付与された音源パラメータが音源レジスタに入力され、楽音生成に使用される。

【0022】そして、この時間フレーム(F-1)において、音源レジスタの音源パラメータを用いて波形データ(WAVE)を生成し、Wバッファ(波形バッファ)に書き込む。このステップS2の実行により、使用された音源パラメータは、Pバッファに記憶しておく必要がなくなり、使用された音源パラメータが記憶されていた領域を解放し、Pバッファには未使用領域(空き)ができる。

【0023】S3においては、再生デバイスが処理を行う。再生デバイスは、CODEC、CODECドライ

バ、サウンドシステム等で構成される。ここでいうCODECとは、音声インタフェース用のLSIであり、内部にA/D変換器、D/A変換器、サンプリング周期発生器、波形圧縮伸張回路、DMAC（ダイレクトアクセスメモリコントローラ）等を備えた半導体チップであり、A/D変換やD/A変換等の機能は有するものの、それだけでは波形データの録音や再生はできない。CODECドライバは、ソフト音源等の他のソフトウェアの下で、上述したCODECを制御あるいは利用して、入力波形をRAM上に録音したり、RAM上の波形データを再生したりするソフトウェアである。なお、CODECドライバが、CODECのLSIの外部に用意されたDMACを利用する場合もある。その場合は、このDMACも再生デバイスに含まれる。

【0024】波形データを生成するプログラムは、波形データをCODECドライバに渡すことにより、CODECを使用した波形再生処理、すなわち、波形データをサンプリング周期毎に1サンプルずつD/A変換器に転送する処理を委託する。CODECドライバに制御されたDMACが、Wバッファに記憶された波形データをサンプリング周期毎に1サンプルずつ読み出し、サウンドシステムを通して波形を出力する。

【0025】一例として、CODECドライバは、DMABバッファとして、1フレーム分のバッファメモリを2個用意している。CODECドライバがこのバッファメモリの一方に記憶された波形データを再生中に、Wバッファに記憶された1フレーム分の波形データが他方のバッファメモリに書き込まれる。一方のバッファメモリから波形データの再生が終了すると、CODECドライバは、他方のバッファメモリの再生を引き続き行い、その間に、Wバッファに記憶された波形データを一方のバッファメモリに書き込む。これを繰り返すことにより、CODECで継続的に波形データが再生される。なお、CODECドライバが1フレーム分のバッファメモリを2個用意する代わりに、上述したWバッファを2個用意して、2つのWバッファに記憶されている波形データを交互に再生し、処理Bは、CODECドライバが1つのWバッファ上の波形データを再生している間に、他方のバッファ上にその次に再生すべき波形データを生成するようにしてもよい。

【0026】また、S1のステップにおいて、発音割当をすることができる。この段階で発音割当を行う場合には、CPUは、ノートオン時に何チャンネルが使用されて発音されているかという発音状況を認識しておく必要がある。ノートオフ時には、ノートオフの音高データに対応する発音中のチャンネル番号を見つけ、このチャンネルの番号の音源レジスタにノートオフを書込むだけでよい。

【0027】この実施の形態においては、曲データから波形データを生成する処理を、前記波形データの再生の進行に応じて処理をする処理Bと、これとは独立して実

行することのできる処理Aとに分けている。そして、処理Aと処理Bとの連携は、具体的には、曲データに含まれたデュレーション情報という時間データを用いて、音源パラメータにも時間情報を付加しておくことにより実現している。処理Aと処理Bとに分けることができたため、処理Aを、Pバッファの記憶容量の範囲で先行して実行することができ、曲データから波形データを再生する処理の負荷が時間的に平均化される。

【0028】上述したPバッファの記憶容量は、必要以上に大きくしなくてもよい。Pバッファの記憶容量は、CPUの処理能力等に応じて最適容量が決まるため、使用されているCPUの種類を検出して自動的にPバッファに割り当てる記憶容量を設定するようにしてもよい。あるいは、ユーザが割り当てる記憶容量を任意に設定するようにしてもよい。

【0029】上述した説明では、曲データの読み出しから波形データの再生までを、処理の流れに沿って説明したが、実際に実行される動作としては、CODECによる波形データの再生処理が「主」であり優先処理され、波形データの再生処理に応じて、処理Bの波形データ生成が「従」で実行される。

【0030】図3は、楽音発生機能を有するパーソナルコンピュータのハードウェア構成の概要図である。21はCPUバス、22はハードディスク、23はFDD、CD-ROM、MOなどのリムーバブルディスク、24はCRT、LCDなどの表示器、25はキーボードおよびマウス、26はCODEC、27はサウンドシステム、28はMIDIインタフェース、29はタイマ、30はCPU、31はROM、32はRAM、33はネットワークインタフェースである。図1、図2に示した第1の実施の形態および図10等を参照して後述する第2の実施の形態に共通して用いることのできる構成を示す。

【0031】通常のパーソナルコンピュータの基本構成にCODEC26、サウンドシステム27が取り付けられたコンピュータであって、さらに、そのコンピュータの基本動作を制御するオペレーティングシステムに、波形再生機能を有するCODECドライバが組み込まれていればソフト音源を実行することができる。ソフト音源を実行するプログラムは、例えば、ハードディスク22に格納されており、RAM32にロードされ実行される。曲データは、例えば、リムーバブルディスク23にあらかじめ記憶されるか、ネットワークインタフェース33を介して外部サーバなどから供給されてハードディスク22にあらかじめ記憶される。

【0032】図1に示したPバッファ、Wバッファは、RAM32上に設けられる。CODEC26は、この中または別に取り付けられたDMAC（ダイレクトメモリアクセスコントローラ）により、RAM32からデータを取り出す。外付けのMIDI機器に接続する場合にはMIDIインタフェース28が、外部サーバに接続する

場合にはネットワークインタフェース33が必要である。CPU30は単一のCPUであるが、マルチCPUであってもよい。

【0033】図4は、本発明の第1の実施の形態の楽音発生方法のメインフローチャートである。このフローは、表示器上で所望の楽曲のMIDIファイル名をクリックするなどして選択すると、ソフト音源プログラムが起動して開始される。S41においては、全ての発音チャンネルのための音源レジスタを無発音状態にする。また、ソフト音源の再生のために、再生デバイスであるCODECドライバ、CODEC26、サウンドシステム27を初期設定し、波形再生ソフトを起動する。S42においては、起動要因チェックが行われ、S43において、起動要因がない場合にはS42に処理を戻し、起動要因がある場合には、S44に処理を進め、起動要因の分析が行われる。

【0034】S44においては、①Pバッファに所定量以上の未使用領域（空き）ができたときには、S45の処理Aに進み、曲データから次に処理すべきMIDIデータを取り出し音源パラメータ（PARM）に変換し、Pバッファに書き込む。②1つの時間フレームの波形再生が完了したときには、S46の処理Bに進み、Pバッファに記憶された演奏タイミング情報付きの音源パラメータに基づいて、次の時間フレームの波形を生成し、Wバッファに書き込む。③音色選択操作、アルゴリズム選択操作等のその他の要求があるときには、S47に処理を進め要求に応じた音源動作の設定を行う。④ソフト音源の終了要求があるときには、S48に処理を進めソフト音源のプログラムの終了処理をする。図3に示したCPU30は、これらの起動要因を絶えずチェックしており、S44において起動要因を判定して具体的な処理を開始する。

【0035】これらの処理の中で最も重要なものは、波形生成をする処理Bである。また、音色の途中変更等は、聴感上処理の遅れがさほど問題にならない。これらの点を考慮して、マルチタスク処理において処理を優先する順位は、②>①>③>④とする。なお、Pバッファ、Wバッファとも、各バッファに記憶されたデータの内、処理の終わった使用済みのデータが記憶されていた領域は、未使用領域として開放される。

【0036】ここで、上述した処理Bの起動要因②の変形例について説明する。上述した説明では、1つの時間フレームの波形再生が完了したときを起動要因とした。その代わりに、CODECドライバが読むCODECのバッファ領域があるが、この領域がある程度空いて所定量以上の未使用領域ができたときを処理Bの起動要因②とすることもできる。

【0037】あるいは、1つの時間フレーム以下の時間間隔で中程度のレベルの割り込みをかける波形生成トリガを頻繁に出しておき、うまく割り込みがかかったとき

を処理Bの起動要因②とし、先に生成できなかった波形があっても波形生成の遅れを回復できるようにする。割り込みのかかり方に変動が多い環境でも音源を安定して動作させることができる。

【0038】なお、CODECのバッファは、通常2フレーム分のバッファである場合が多いが、バッファサイズはこれ以上にしてもよい。いずれの変形例でも、起動要因②で起動される処理Bでは、その時点でのCODECのバッファの未使用領域の大きさに応じた所定量の波形を生成すればよい。上述した起動要因の変形例において、Wバッファは必ずしも必要としない。

【0039】図5は、図4に示した処理Aのフローチャートである。S51においては、曲データから、次の演奏イベントの演奏データおよび演奏タイミングを取り出す。具体的には、曲データから既に読み終わった演奏イベントをポインタで示すようにしておくことによって、次のものを取り出す。演奏データはMIDIデータである。演奏タイミングは、MIDIデータに含まれているデュレーション情報または、これから求めた時刻情報である。

【0040】S52においては、各チャンネル（ch）のエンベロープを、この演奏タイミングまで計算し、Eバッファ（エンベロープバッファ）に書き込んでおく。このEバッファは、図1に示したPバッファ、Wバッファとは全く別のものである。従来のハードウェア音源では、エンベロープの読み出し機能を有しており、各発音チャンネルのエンベロープのレベルを見て発音チャンネルの割当を行っていた。その処理ルーチンをそのままソフト音源に流用する場合には、まずエンベロープを生成しなければならない。演奏イベントが発生していないときでも、発音中はエンベロープのレベルが必要であるから、例えば、1つの時間フレームごとに1個のエンベロープのレベルを作成してEバッファに入れておき、その時間フレーム内のエンベロープのレベルは、補間処理によって生成するようにする。

【0041】S53においては、演奏データを音源パラメータに変換するとともに、新規発音の場合には、上述したエンベロープのレベルに基づいてここで発音割当てをする。S54においては、この音源パラメータを、上述した演奏タイミングとともにPバッファに書き込む。

【0042】なお、S52を削除し、音源パラメータ発生時にエンベロープの作成をしないようにすることも可能である。この場合、音源パラメータ作成の段階では、発音チャンネル割り当てをせずに未定にしておき、後述する処理Bの波形生成時に発音チャンネル割り当てを行う。ただし、この場合、演奏イベントが重なりと処理が重くなるため、音源パラメータ作成の段階で発音チャンネル割当を行う方が望ましい。あるいは、音源パラメータ作成の段階で発音チャンネル割当を行うものの、エンベロープレベルを用いずに、例えば、後着優先で発音割り当てを

する、すなわち、後から入力された発音を優先して行うようにすることも可能である。

【0043】図6は、図4に示した処理Bのフローチャートである。処理Bは波形を生成するときに起動される。S61においては、波形生成量が決定される。図2に示したように、1つの時間フレームごとに割り込みが入るときには、この波形生成量は1フレームに固定される。先に説明した図4の起動要因②の変形例の場合には、CODECのバッファの未使用領域の大きさに応じて波形生成量が決定される。

【0044】S62において、波形生成する範囲に未処理の演奏イベントがあるかどうかを判定する。この判定は、曲データから既に読み終わって処理Aが実行された演奏イベントを示すポイントを調べ、その次の演奏イベントの演奏タイミングが、波形生成しようとしている範囲よりも後の時刻であるかどうかで行う。普通は、演奏イベントが発生する間隔が数ms～数十msの時間フレームに対して長いため、波形生成しようとしている範囲内にある演奏イベントは、先行して処理Aを終えているはずである。しかし、曲データ中には、この波形生成する範囲内に、まだ処理Aを終えていない演奏イベントが残っている可能性がある。

【0045】図7は、波形生成する範囲に未処理のイベントがある場合の第1の例のタイミング図である。図中、図14、図2と同様な部分には同じ符号を付して説明を省略する。図7に示すように、時間フレーム(F)で波形再生する必要のある演奏イベントが残っていたことを、時間フレーム(F-1)において判定したときには、図6のS63に処理を進め、未処理の演奏イベントの処理Aを、時間フレーム(F-1)の音源パラメータ作成期間12において実行し、S62に処理を戻す。S62において、波形生成する範囲に未処理の演奏イベントがないと判定したときには、S64に処理を進める。S63は繰り返し実行される場合がある。

【0046】S64においては、波形生成すべき時間フレーム内に、Pバッファに記憶された音源パラメータの演奏タイミングが入る場合には、1フレーム分の波形データの生成の途中で、生成位置がこの演奏タイミングに一致する時点でPバッファに記憶された音源パラメータで音源レジスタを更新しつつ、この音源レジスタに記憶された音源パラメータに基づいて波形データを生成する。例えば、ノートオンの場合には、発音に必要な音源パラメータを音源レジスタの1つのチャンネルに転送するとともに、ノートオン/ノートオフを示すレジスタをオンにして発音状態にし、ノートオフの場合には、単にノートオン/ノートオフを示すレジスタをオフに書き換える。これとともに、音源レジスタに記憶された音源パラメータに基づいて、生成量分、例えば、1フレーム分の波形を生成し、Wバッファに書き込む。

【0047】波形生成に際しては、図5に示した処理A

のS52においてEバッファに書き込んだエンベロープも使用する。図5に示した処理AのS52においてエンベロープを作らない場合には、ここでエンベロープを作成しつつ波形生成を行う。S62において波形データを生成する範囲に未処理の演奏イベントなしと判断されたにもかかわらず、波形データを生成する時間フレームに含まれた演奏イベントのエンベロープが、Eバッファにまだ作られていない場合がある。すなわち、処理Aは、波形データを生成する時間フレームに含まれる演奏イベントまで完了しているが、その次の演奏イベントの処理Aは未処理である場合、つまり、波形データ生成に必要なエンベロープ波形のうちで、処理済みの最終イベントの所までは処理Aでエンベロープが作成済みであるが、その後の部分が未作成であるという場合である。補間処理では、この時間フレームに含まれる演奏イベントのエンベロープを作ることができない。このような場合、このS64のステップで、波形データを生成する時間フレームについて、エンベロープを作成する必要がある。S65においては、生成してWバッファに書き込んだ1フレーム分の波形データをCODECドライバに渡す。

【0048】ここで、S62の変更例を説明する。S62において、処理Aが、波形生成すべき時間フレームよりも後の演奏イベントを少なくとも1つは処理済みであるかどうかを判定し、処理を終えていない場合はS63に処理を進め、少なくとも1つの処理が終わってからS64に処理を進める。このようにすれば、処理Aにより、波形生成すべき時間フレームよりも後の演奏イベント13の所までエンベロープが作成されるので、S64の波形生成ではエンベロープの作成をしなくてもよい。

【0049】図8は、波形生成する範囲に未処理のイベントがある場合の第2の例のタイミング図である。図中、図14、図2と同様な部分には同じ符号を付して説明を省略する。図8に示すように、時間フレーム(F-1)において音源パラメータに変換されていない未処理の演奏イベントが、例えば、波形生成する時間フレーム(F)の範囲よりも後の時間フレーム(F+2)で波形生成するものであると判定したときには、図6に示したS62では、S64に処理を進める。これに対して、S62の変更例では、S63に処理を進めて、この未処理の演奏イベントの曲データを音源パラメータに変換した上で、処理Bを実行する。このようにして、波形生成する範囲よりも後の演奏タイミングの演奏イベントの処理Aが必ず実行された後に波形処理を行うようにすることができる。

【0050】図9は、DMACのフローチャートである。図3に示したCODECによる波形再生プログラムの処理の流れを示すものである。S71においては、サンプリング周期でCODECが発生するサンプル要求割り込み(ハードウェア割り込み)に応じて、CODECの波形バッファであるDMAバッファ(DMAB)は、

波形データを1サンプルずつCODECに送出する。S72においては、転送サンプル数pの値を1だけ増加させてS71の処理を繰り返し実行させる。

【0051】転送サンプル数pを、例えば8ビットで表わし、225までカウントすれば次に0に戻るようになる。CODECは、転送サンプル数pをカウントして、DMAバッファのサイズの半分の数のサンプルが転送される毎、すなわち、転送サンプル数p=127およびp=255のときに、1フレームの波形再生が完了したとしてハードウェア割り込みを実行する。このハードウェア割り込みが、処理Bの開始トリガ、すなわち、図4に示したS44における②の起動要因となっている。

【0052】処理Bの波形生成処理では、DMAバッファの半分に相当する1フレーム分のサンプルを、Wバッファ上にまとめて一気に生成し、Wバッファの波形データはDMAバッファに転送される。上述したDMAバッファのサイズは、2つの時間フレーム分であるが、このサイズは、DMAコントローラ(DMAC)のアドレス設定を変えることで任意に変更可能である。

【0053】図10は、本発明の第2の実施の形態の楽音発生方法の処理の流れを表わす説明図である。ストリーム再生の場合を示すものである。S81においては、ネットワークからマルチメディアデータ(MMD)を受信し、これを図3に示したRAM32上の受信バッファに書き込む。この受信バッファの領域は、例えば1曲分のマルチメディアデータを書き込むことができる大きさが確保されている。なお、マルチメディアデータとは、MIDIデータのほかに、これと同期する歌詞データや画像データ等を含むデータである。

【0054】S82においては、受信バッファから読み出したマルチメディアデータ(MMD)からMIDIデータからなる曲データ(MIDI)を抽出し、Mバッファ(MIDIバッファ)に書き込む。抽出されたMIDIデータにも、図1を参照して説明した曲データと同様に、演奏イベントの時間間隔を表わすデュレーション情報が含まれている。このデュレーション情報から演奏タイミングを示す時刻情報データへの変換を行ない、演奏イベントのMIDIデータに付け加え、Mバッファに曲データ(MIDI)が記憶される。ただし、S82は、Mバッファに未使用領域(空き)があり、かつ、受信バッファに未処理のマルチメディアデータ(MMD)があれば実行する。

【0055】S83以降については、図1に示した曲ファイル再生の処理の流れの場合のS1~S3とほぼ同様であるが、若干異なる部分もある。S83の処理Aにおいては、Pバッファに未使用領域があり、かつ、Mバッファに曲データ(MIDI)があれば実行する。また、このステップの実行により、使用された曲データは、Mバッファに記憶しておく必要がなくなり、使用された音源パラメータが記憶されていた領域を解放するため、M

バッファに未使用領域(空き)ができる。S84については、図13を参照して後述するように、波形生成する範囲に未処理の演奏イベントがある場合の処理が異なる。図1に示したS1で作成される音源パラメータと同様に、S83において作成される音源パラメータも演奏タイミングを示す時間情報付きの音源パラメータである。

【0056】この実施の形態においては、受信データから波形データを生成する処理を、波形データの再生の進行に応じて処理をする処理Bと、これとは独立して実行することのできる処理Aおよび曲データ抽出処理とに分けている。そして、これらの処理間の連携は、具体的には、受信されたマルチメディアデータ(MMD)中の曲データ(MIDI)に含まれたデュレーション情報という時間データを用いて、曲データ(MIDI)、音源パラメータ(PARM)にも時間情報を付加しておくことにより実現している。

【0057】処理Bを曲データ抽出処理および処理Aから分けることができたため、曲データ抽出処理およびパラメータ変換処理を、Mバッファ、Pバッファの記憶容量の範囲で先行して実行することができ、受信データから波形データを再生する処理の負荷が時間的に平均化される。

【0058】さらに、Mバッファに曲データが存在する場合にパラメータに変換する処理を行うことができるため、曲データの全部のロードが完了するのを待つことなく、曲の再生を開始することができる。なお、受信バッファをある程度大きな容量にして、上述したように、例えば1曲分のマルチメディアデータを書き込むことができる大きさを確保しておき、受信したマルチメディアデータを書き換えずに残しておくようにした場合には、受信バッファ上に残されたデータを再び再生することができる。あるいは、この受信バッファを書き込みバッファとして、受信したマルチメディアデータをそのまま図3に示したハードディスク22やリムーバブルディスク23に書き込むようにしてもよい。また、受信バッファではなくMバッファをある程度大きな容量にして、曲データを書き換えずに残しておくようにした場合には、曲データのみを再び再生することができる。

【0059】図11は、本発明の第2の実施の形態の楽音発生方法のメインフローチャートである。このフローを実現するプログラムは、例えば、ウェブブラウザに組み込まれるプラグインソフトとして実現される。スタンダードMIDIファイル(SMF)のような曲データ、または、このような曲データを含むマルチメディアデータ(MMD)がサーバに記憶されている位置を示す「タグ」データがホームページに埋め込まれていて、この「タグ」データに応じてダウンロードが指示されたときに、本プログラムが起動され、ダウンロードするだけでなくマルチメディアデータの再生まで行う。

【0060】S91においては、全ての発音チャンネルのための音源レジスタを無発音状態にし、また、CODE Cを初期設定し波形再生ソフトを起動する。S92においては、起動要因チェックが行われ、S93においては、起動要因がない場合にS92に処理を戻し、起動要因がある場合には、S94に進み起動要因の分析が行われる。

【0061】S94においては、①ネットワークからマルチメディアデータ(MMD)の受信があるときには、S95に処理を進め、受信処理を行い、マルチメディアデータ(MMD)を受信バッファに書き込む。②Mバッファに未使用領域(空き)があり、かつ、受信バッファにマルチメディアデータ(MMD)があるときには、S96に処理を進め、受信バッファからマルチメディアデータ(MMD)を読み出し、曲データ(MIDI)抽出処理を行う。③Pバッファに未使用領域があり、かつ、Mバッファに曲データ(MIDI)があるときには、S97の処理Aに進み、Mバッファから次に処理すべき曲データ(MIDI)を取り出し、音源パラメータ(PARM)に変換しPバッファに書き込む。

【0062】④1つの時間フレームの波形再生が完了したときには、S98の処理Bに進み、Pバッファから読み出した演奏タイミング情報付きの音源パラメータに基づいて、次の時間フレームの波形を生成する。⑤音色選択などのスイッチング操作、歌詞、画像の表示の差し替えやスクロールといったマルチメディア処理等、その他の要求があるときには、S99に処理を進め、これらの要求に応じた各種の設定を行う。⑥音源ソフトの終了要求があるときには、S100に処理を進めソフト音源のプログラムの終了処理をする。図3に示したCPU30は、これらの起動要因を絶えずチェックしており、S94において起動要因を判定して具体的な処理を開始する。

【0063】ダウンロードの開始時は、最初に、①の受信処理と②の曲データ抽出処理とをスタートさせ、所定量または所定時間分の曲データがMバッファにたまった後に、はじめて③の処理Aと④の処理Bというソフト音源の波形生成処理を開始するようにする。あるいは、所定量または所定時間分のマルチメディアデータが受信バッファにたまった後に、上述した波形生成処理を開始するようにしてもよい。さらにまた、③の処理Aも①の受信処理や②の曲データ抽出処理と同時に、④の処理Bよりも先にスタートさせ、所定量または所定時間分の音源パラメータがPバッファにたまった後に④の処理Bを開始するようにしてもよい。このようにして、各バッファにある程度データがたまってから波形生成を開始するので、再生開始直後の動作を安定して行うことができる。なお、上述した所定量または所定時間は、ネットワークの転送レート、回線品質等によって必要量が異なるから、ユーザが任意に設定可能にしておく。処理の優先順

位は、①>④>⑤のマルチメディア処理>②>③>⑤のスイッチング操作>⑥という優先順位でマルチタスク処理を行う。なお、Mバッファ、Pバッファ、Wバッファとも、各バッファに記憶されたデータの内、処理の終わった使用済みのデータが記憶されていた領域は、未使用領域として開放される。

【0064】図12は、図11に示した曲データ抽出処理のフローチャートである。S111においては、受信バッファから次の1組のマルチメディアデータ(MMD)を取り出し、S112においては、マルチメディアデータ(MMD)の種類を判定し、曲データ(MIDI)であれば、S113に処理を進め、その他のマルチメディアデータであれば、S114に処理を進める。S113においては、この1組の曲データをMバッファに書き込み、S114においては、この1組のマルチメディアデータを、その種類に対応したバッファ、例えば、画像データならGバッファ、歌詞データならKバッファ等、に書き込み、記憶されたデータは、それぞれ、それらのデータの再生タイミングにおいて、図11に示したS99のその他の処理において、その表示のための処理が行われる。

【0065】マルチメディアデータに含まれる歌詞データや画像データは、歌詞や画像データをディスプレイに表示する時刻を示すタイムスタンプ、タイムコードのような時刻情報を含み、この時刻情報が付いた状態でそれぞれに対応したバッファに書き込まれ、表示の際には、この時刻情報と、ソフト音源で生成される波形データの再生タイミング、すなわち、演奏イベントの演奏タイミングを示す時刻情報とを用いて、波形データの再生と歌詞データ、画像データの表示とを正確に同期させることが容易に行える。また、演奏イベントの演奏タイミングを時間間隔情報のままで音源パラメータに付けてパラメータバッファに記憶しておく場合にも、この時間間隔情報を加算して常に時刻情報を得ていれば同様に同期させることができる。

【0066】図13は、図11に示した処理Bのフローチャートである。S121においては、波形生成量を決定し、S122においては、波形生成する範囲に未処理の演奏イベントが有るかどうかを判定する。この判定は、曲データから既に読み終わって処理Aが実行された演奏イベントを示すポイントを調べ、その次の演奏イベントの演奏タイミングが、波形生成しようとしている範囲よりも後の時刻であるかどうかで行う。未処理の演奏イベントがない場合には、S123に処理を進め、有る場合にはS124に処理を進める。

【0067】S123においては、波形生成すべき時間フレーム内に、Pバッファに記憶された音源パラメータの演奏タイミングが入る場合には、1フレーム分の波形データの生成の途中で、生成位置がこの演奏タイミングに一致する時点でPバッファに記憶された音源パラメー

タで音源レジスタを更新しつつ、この音源レジスタに記憶された音源パラメータに基づいて波形データを生成する。そして、生成量分、例えば、1フレーム分の波形をWバッファに書き込み、S125においては、生成してWバッファに書き込んだ1フレーム分の波形データをCODECドライバに渡す。

【0068】S124は、波形生成に必要な音源パラメータが用意できていないときの処理であり、波形生成する範囲に未処理の曲データが有るかどうかを判定する。受信データから既に読み終わって曲データが抽出された演奏イベントを示すポイントを調べ、このポイントの次の演奏イベントの演奏タイミングが波形生成しようとしている範囲よりも後の時刻であるかどうかで行う。未処理の曲データがない場合にはS126に処理を進め、有る場合には、S127に処理を進める。S126においては、未処理の演奏イベントの処理Aを実行し、S122に処理を戻す。

【0069】S127は、曲データが用意できていないときの処理であり、受信バッファに未処理の受信データが有るかどうかを、曲データ抽出処理を行った最後の受信データが最新の受信データであるかどうかを調べて判定し、未処理の受信データが有る場合には、S128に処理を進め、ない場合には、S129に処理を進める。S128においては、曲データ抽出処理を実行し、S124に処理を戻す。

【0070】S129は、必要な受信データが未到着のときのエラー処理である。エラー処理では、音源レジスタの現在値のみに基づいて波形を生成し、S125に処理を進めることにより、既に鳴っている音を継続するように処理する。あるいは、既に鳴っている音を徐々に減衰するように制御してもよい。

【0071】なお、S122の処理としては、図6に示した曲データ再生の場合の処理Bにおける、S62の変更例と同様な処理をすることもできる。すなわち、波形生成する範囲よりも後の時刻の演奏イベントの少なくとも1つについて、処理Aが必ず実行された後に、波形データ生成処理を行うようにすることができる。同様な考えで、S124においても、波形生成する範囲よりも後の演奏タイミングの演奏イベントの少なくとも1つについて、曲データ抽出処理が必ず実行された後に、パラメータ変換および波形データ生成処理を行うように変更することができる。

【0072】この変更例であれば、図6に示した曲データ再生の場合の処理BにおけるS64の波形生成の場合と同様に、今回波形生成する時間フレームについて、処理Aの中でエンベロープ生成が終わっていることが保証できるので、S123において、エンベロープを作成する必要がない。逆に、S122を変更しない場合には、S123において、エンベロープを作成する必要がある場合がある。

【0073】図10ないし図13を参照して説明した第2の実施の形態は、ネットワークからデータを受信する場合に限らず、例えば、外部記憶媒体からマルチメディアデータを転送する場合に、転送をしつつ再生を行う場合や、無線伝送や光伝送でデータを受信しながら再生を行う場合にも適用することができる。

【0074】上述した説明では、楽音発生処理に極力エラーが生じないようにしたため、例えば、図6のS62、S63のように、処理Bの中で処理Aに戻るフローを設けた。しかし、CPUパワーが大きい場合など、例外的な状況が発生するおそれが少ない環境で使用する場合には、このような処理を戻すフローを省略することも可能である。図13のS122、S124、S126、S127、S128も同様な処理を戻すフローである。また、図1ないし図9を参照して説明した曲ファイルの再生においては、エラー処理のフローを設けなかったが、図10ないし図13を参照して説明したストリーム再生の場合の図13のS129のようなエラー処理を行うようにフローを変更することもできる。

【0075】上述した説明では、曲データに含ませる演奏タイミングのデータとしてデュレーション情報を用いた場合について説明したが、その代わりに、曲の先頭などからの絶対時間を表す時刻情報を用いてもよい。この場合、デュレーション情報を時刻情報データに変換することなく、Pバッファには、この時刻情報を付加した音源パラメータが書き込まれる。また、汎用のパーソナルコンピュータを前提にして説明したが、音源専用のコンピュータを用いても実現することができる。ソフト音源に適用した例について説明したが、音源用のハードウェアを用いたものに適用することも可能である。楽音は、楽器音に限らず音声、擬音等の広い意味での可聴音を意味する。電子楽器の音源だけでなく、ゲームやカラオケなど娯楽装置、テレビジョンなどの各種家電機器の楽音発生方法として用いることができる。

【0076】

【発明の効果】上述した説明から明らかなように、負荷が分散され実質的な処理負担が小さくなる楽音発生方法を実現できるという効果がある。波形データを再生デバイスの再生タイミングに先行して生成する場合に比べて、バッファサイズを小さくできるので、使用するRAMの容量を抑えることができるという効果がある。また、ネットワークから曲データをロードして波形データを再生する場合に、曲データの全部のロードが完了するのを待つことなく、曲の再生を開始できるという効果がある。

【図面の簡単な説明】

【図1】 本発明の第1の実施の形態の楽音発生方法の処理の流れを表わす説明図である。

【図2】 本発明の第1の実施の形態の楽音発生方法のタイミング図である。

【図3】 楽音発生機能を有するパーソナルコンピュータのハードウェア構成の概要図である。

【図4】 本発明の第1の実施の形態の楽音発生方法のメインフローチャートである。

【図5】 図4に示した処理Aのフローチャートである。

【図6】 図4に示した処理Bのフローチャートである。

【図7】 波形生成する範囲に未処理のイベントがある場合の第1の例のタイミング図である。

【図8】 波形生成する範囲に未処理のイベントがある場合の第2の例のタイミング図である。

【図9】 DMACのフローチャートである。

【図10】 本発明の第2の実施の形態の楽音発生方法の処理の流れを表わす説明図である。

【図11】 本発明の第2の実施の形態の楽音発生方法のメインフローチャートである。

【図12】 図11に示した曲データ抽出処理のフローチャートである。

【図13】 図11に示した処理Bのフローチャートである。

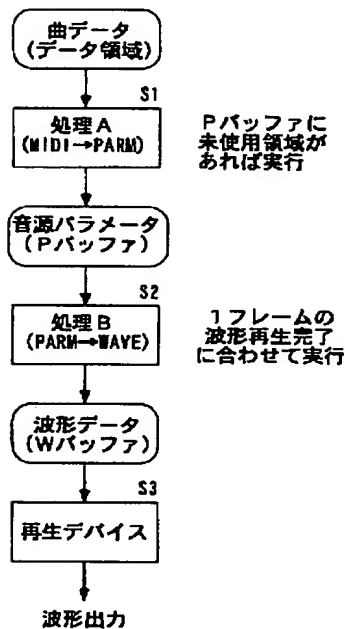
【図14】 従来の楽音発生方法の動作説明図である。

【符号の説明】

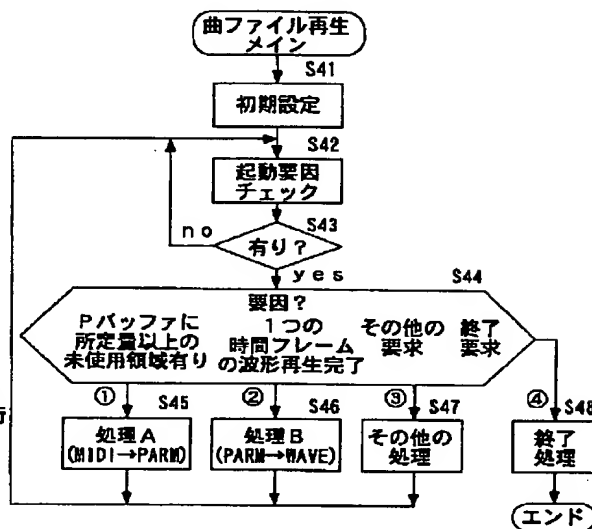
11…曲データ、12…音源パラメータ作成期間、13…演奏タイミング、14…波形データ生成期間、121…MIDIデータの入力タイミング

【図1】

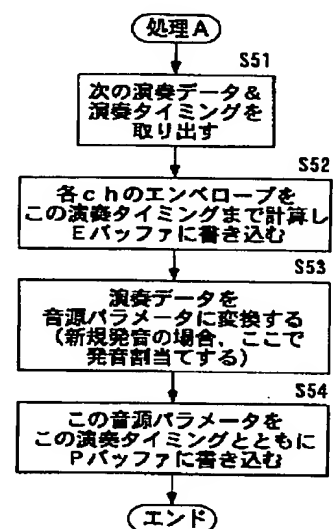
曲ファイルの再生



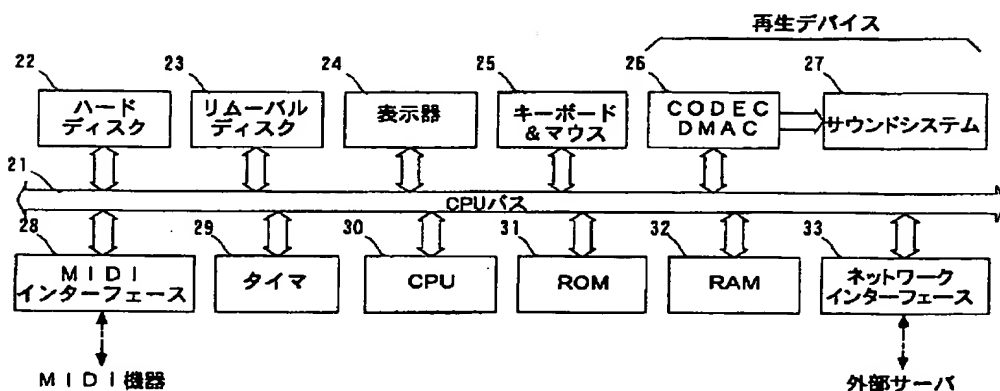
【図4】



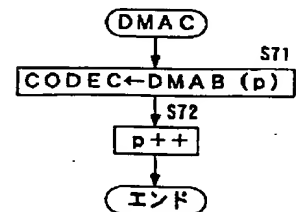
【図5】



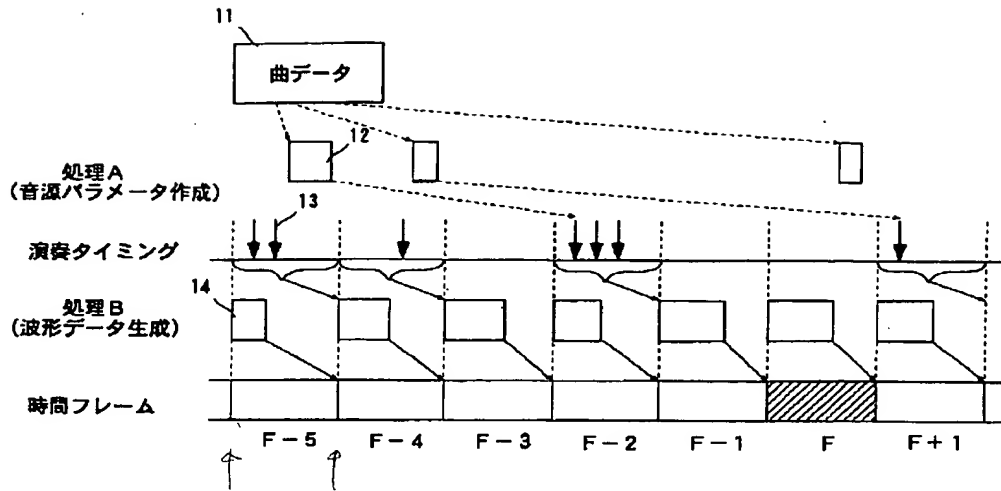
【図3】



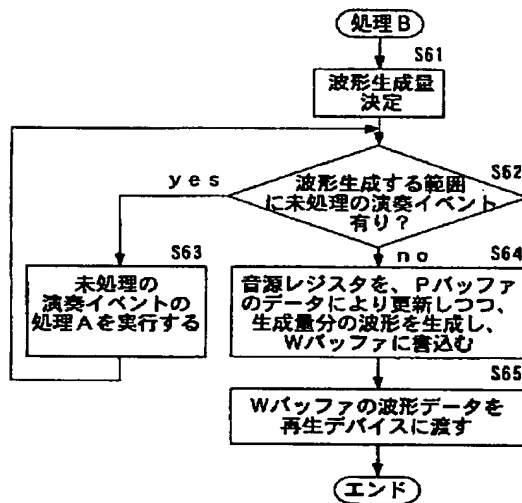
【図9】



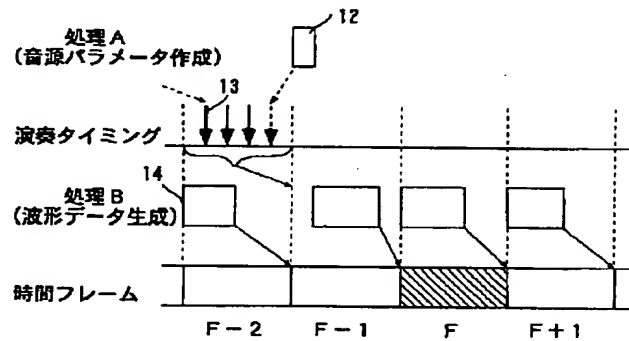
【図2】



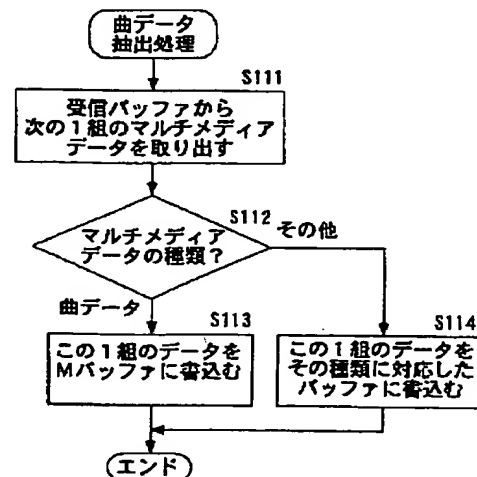
【図6】



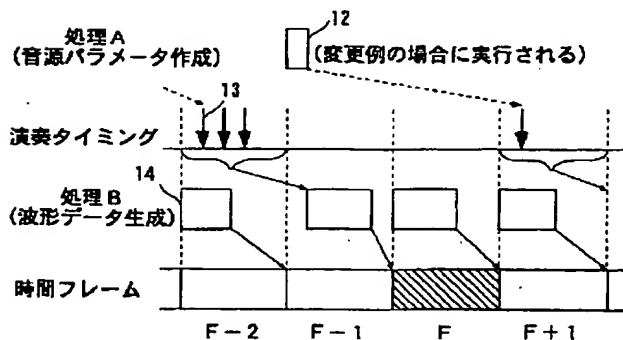
【図7】



【図12】

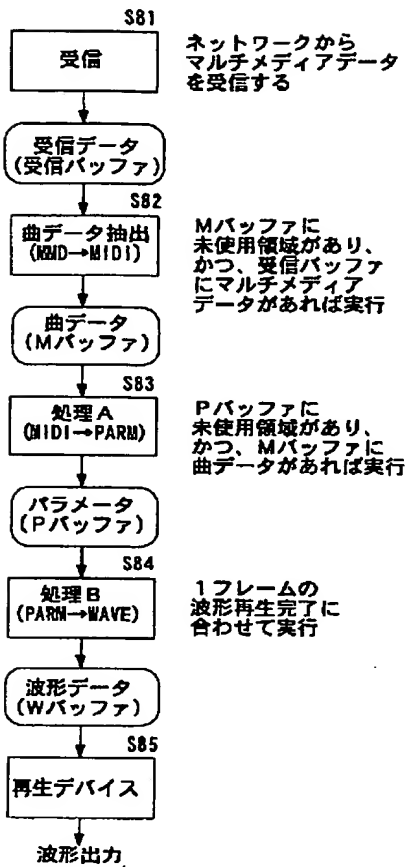


【図8】

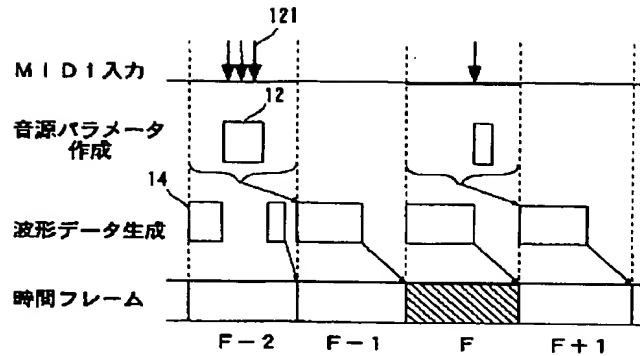


【図10】

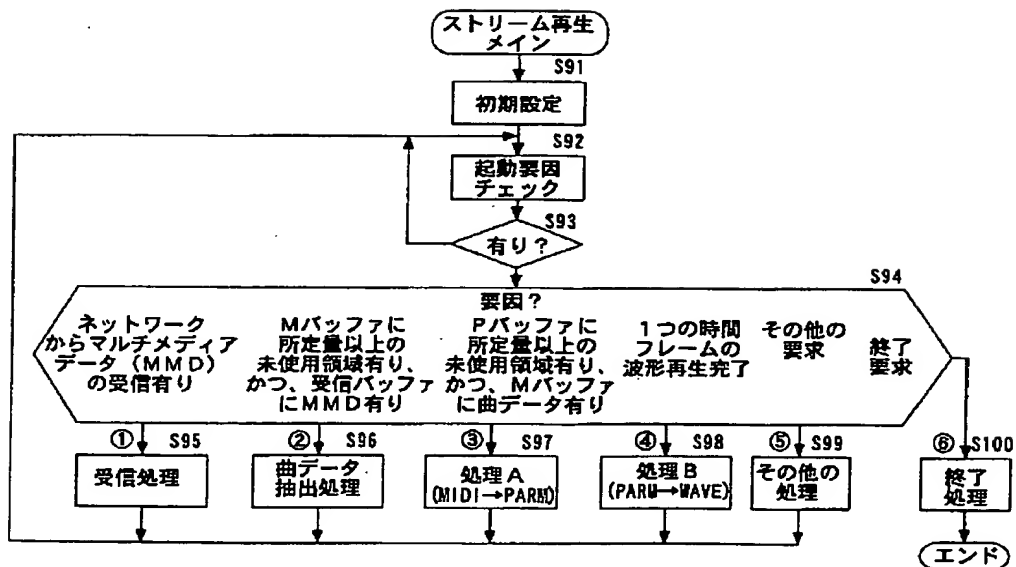
ストリーム再生



【図14】



【図11】



【図13】

